Model_Transfer_learning

July 6, 2024

0.1 Mounting Google Drive

[3]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

0.2 Import Libraries

[2]:
```
!pip install tensorflow-addons
import os
import cv2
import numpy as np
import tensorflow as tf
import shutil
import random
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
from tensorflow import keras
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.applications import InceptionV3, VGG16
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Conv2D,␣
↪MaxPooling2D, Flatten, BatchNormalization, Dropout, Input, Lambda
from tensorflow.keras.optimizers import Adam
import tensorflow_addons as tfa
from sklearn.metrics import confusion_matrix
import seaborn as sns
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau,␣
↪EarlyStopping
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications.inception_v3 import preprocess_input
from glob import glob
```

```
Collecting tensorflow-addons
```

```
  Downloading tensorflow_addons-0.23.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (611 kB)
                           611.8/611.8
kB 7.2 MB/s eta 0:00:00
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from tensorflow-addons) (24.1)
Collecting typeguard<3.0.0,>=2.7 (from tensorflow-addons)
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Installing collected packages: typeguard, tensorflow-addons
Successfully installed tensorflow-addons-0.23.0 typeguard-2.13.3

/usr/local/lib/python3.10/dist-
packages/tensorflow_addons/utils/tfa_eol_msg.py:23: UserWarning:

TensorFlow Addons (TFA) has ended development and introduction of new features.
TFA has entered a minimal maintenance and release mode until a planned end of
life in May 2024.
Please modify downstream libraries to take dependencies from other repositories
in our TensorFlow community (e.g. Keras, Keras-CV, and Keras-NLP).

For more information see: https://github.com/tensorflow/addons/issues/2807

  warnings.warn(
```

0.3 Resizing Images in Subfolders

[3]:
```python
main_folder = "/content/drive/MyDrive/Car_image_classification_CNN_model/Data"

# Function to resize images in a directory to the specified size using cv2
def resize_images_in_folder(folder_path, target_size):
    for filename in os.listdir(folder_path):
        file_path = os.path.join(folder_path, filename)
        if os.path.isfile(file_path) and filename.endswith('.jpg'):
            try:
                img = cv2.imread(file_path)
                img_resized = cv2.resize(img, target_size)
                cv2.imwrite(file_path, img_resized)

            except Exception as e:
                print(f"Error resizing image {file_path}: {e}")

# Iterate through subfolders in the main folder and resize images
for subfolder in os.listdir(main_folder):
    subfolder_path = os.path.join(main_folder, subfolder)
    if os.path.isdir(subfolder_path):
        resize_images_in_folder(subfolder_path, (480, 480))
```

0.4 Splitting Data into Train, Validation, and Test Sets

[4]:
```
main_folder = "/content/drive/MyDrive/Car_image_classification_CNN_model/Data"
output_folder = "/content/drive/MyDrive/Car_image_classification_CNN_model/
 ↪split_data"

# Function to create output folders
def create_output_folders(output_folder):
    # Create output folders if they don't exist
    for folder in ['train', 'val', 'test']:
        main_output_folder = os.path.join(output_folder, folder)
        os.makedirs(main_output_folder, exist_ok=True)

        # Create subfolders inside each main output folder
        for subfolder in os.listdir(main_folder):
            subfolder_path = os.path.join(main_output_folder, subfolder)
            os.makedirs(subfolder_path, exist_ok=True)

# Function call for output folders
create_output_folders(output_folder)

print("Output folders created successfully.")

# Function to split images into train, test, and validation sets and save them␣
 ↪in respective folders
def split_data_into_sets(main_folder, output_folder, test_size=0.1, val_size=0.
 ↪1):
    for subfolder in os.listdir(main_folder):
        subfolder_path = os.path.join(main_folder, subfolder)
        if os.path.isdir(subfolder_path):
            image_files = [os.path.join(subfolder_path, file) for file in os.
 ↪listdir(subfolder_path) if file.endswith('.jpg')]
            train_files, test_val_files = train_test_split(image_files,␣
 ↪test_size=test_size + val_size, random_state=42)
            test_files, val_files = train_test_split(test_val_files,␣
 ↪test_size=val_size/(test_size + val_size), random_state=42)

            # Copy images to respective folders
            for file in train_files:
                shutil.copy(file, os.path.join(output_folder, 'train',␣
 ↪subfolder))
            for file in val_files:
                shutil.copy(file, os.path.join(output_folder, 'val', subfolder))
            for file in test_files:
                shutil.copy(file, os.path.join(output_folder, 'test',␣
 ↪subfolder))
```

```
# Function call Split images into train, test, and validation sets and save␣
 ↪them in respective folders
split_data_into_sets(main_folder, output_folder)

print("Images split and saved successfully.")
```

```
Output folders created successfully.
Images split and saved successfully.
```

0.5 Data Paths and Class Count

[5]:
```
train_path = '/content/drive/MyDrive/Car_image_classification_CNN_model/
 ↪split_data/train'
valid_path = '/content/drive/MyDrive/Car_image_classification_CNN_model/
 ↪split_data/val'
test_path  = '/content/drive/MyDrive/Car_image_classification_CNN_model/
 ↪split_data/test'
```

[6]:
```
num_class= glob('/content/drive/MyDrive/Car_image_classification_CNN_model/
 ↪split_data/train/*')
print("number of class:-",len(num_class))
```

```
number of class:- 9
```

0.6 Image Data Generators augmentation

[7]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                  )

val_datagen  = ImageDataGenerator(rescale = 1./255)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

0.7 Creating Training ,Validation and Tes Data Generator

[8]:
```
training_set = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/
 ↪train',
    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
    classes= ['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
```

```
)
```

```
Found 4331 images belonging to 9 classes.
```

[9]:
```
val_set = val_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/val',
    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
)
```

```
Found 1008 images belonging to 9 classes.
```

[10]:
```
test_set = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test',
    target_size=(480, 480),
    batch_size=32,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
)
```

```
Found 991 images belonging to 9 classes.
```

0.8 Visualization of training dataset with the original lables

[11]:
```
training_set = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/
 ↪train',
    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
    )

class_labels = {v: k for k, v in training_set.class_indices.items()}

plt.figure(figsize=(20, 20))
for i in range(25):
    batch_idx = random.randint(0, len(training_set) - 1)
    batch = training_set[batch_idx]
```

```
    image_idx = random.randint(0, len(batch[0]) - 1)

    image = batch[0][image_idx]


    label = batch[1][image_idx]


    class_index = np.argmax(label)


    if np.max(image) <= 1.0:
        image = (image * 255).astype(np.uint8)

    plt.subplot(5, 5, i + 1)
    plt.imshow(image)
    plt.title(f"Label: {class_labels[class_index]}")
    plt.axis("off")

plt.show()
```

Output hidden; open in https://colab.research.google.com to view.

0.9 Visualization of validation dataset with the original lables

[12]:
```
val_set = val_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/val',
    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
     classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
)


class_labels = {v: k for k, v in val_set.class_indices.items()}

plt.figure(figsize=(20, 20))
for i in range(25):
    batch_idx = random.randint(0, len(val_set) - 1)
    batch = val_set[batch_idx]
    image_idx = random.randint(0, len(batch[0]) - 1)

    image = batch[0][image_idx]
```

```
    label = batch[1][image_idx]

    class_index = np.argmax(label)


    if np.max(image) <= 1.0:
        image = (image * 255).astype(np.uint8)

    plt.subplot(5, 5, i + 1)
    plt.imshow(image)
    plt.title(f"Label: {class_labels[class_index]}")
    plt.axis("off")

plt.show()
```

Output hidden; open in https://colab.research.google.com to view.

0.10 Visualization of testing dataset with the original lables

[13]:
```
test_set = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test',
    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
)

class_labels = {v: k for k, v in test_set.class_indices.items()}

plt.figure(figsize=(20, 20))
for i in range(25):
    batch_idx = random.randint(0, len(test_set) - 1)
    batch = test_set[batch_idx]
    image_idx = random.randint(0, len(batch[0]) - 1)

    image = batch[0][image_idx]


    label = batch[1][image_idx]

    class_index = np.argmax(label)


    if np.max(image) <= 1.0:
        image = (image * 255).astype(np.uint8)
```

```
    plt.subplot(5, 5, i + 1)
    plt.imshow(image)
    plt.title(f"Label: {class_labels[class_index]}")
    plt.axis("off")

plt.show()
```

Output hidden; open in https://colab.research.google.com to view.

0.11 InceptionV3 Model Creation and Compilation

[14]:
```
# Importing the InceptionV3  library and adding preprocessing layer to the␣
 ↪front of InceptionV3
IMAGE_SIZE = [480, 480]
inception_base = InceptionV3(input_shape=IMAGE_SIZE + [3], weights='imagenet',␣
 ↪include_top=False)
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applicatio
ns/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [==============================] - 0s 0us/step

[15]:
```
# Keeping the base existing weights
for layer in inception_base.layers:
    layer.trainable = False
```

[16]:
```
x = Flatten()(inception_base.output)
x = Dense(256, activation='relu')(x)
prediction = Dense(len(num_class), activation='softmax')(x) # addded layers
```

[17]:
```
#  model object
model = Model(inputs=inception_base.input, outputs=prediction)
```

[18]:
```
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
 metrics=['accuracy',
             tf.keras.metrics.Precision(name='precision'),
             tf.keras.metrics.Recall(name='recall'),
             tfa.metrics.F1Score(num_classes=len(num_class),␣
 ↪average='weighted', name='f1_score')]
)

model.summary()
```

Model: "model"

```
------------------
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================
==================
 input_1 (InputLayer)        [(None, 480, 480, 3)]     0         []

 conv2d (Conv2D)             (None, 239, 239, 32)      864
['input_1[0][0]']

 batch_normalization (Batch  (None, 239, 239, 32)      96
['conv2d[0][0]']
 Normalization)

 activation (Activation)     (None, 239, 239, 32)      0
['batch_normalization[0][0]']

 conv2d_1 (Conv2D)           (None, 237, 237, 32)      9216
['activation[0][0]']

 batch_normalization_1 (Bat  (None, 237, 237, 32)      96
['conv2d_1[0][0]']
 chNormalization)

 activation_1 (Activation)   (None, 237, 237, 32)      0
['batch_normalization_1[0][0]'
                                                                  ]

 conv2d_2 (Conv2D)           (None, 237, 237, 64)      18432
['activation_1[0][0]']

 batch_normalization_2 (Bat  (None, 237, 237, 64)      192
['conv2d_2[0][0]']
 chNormalization)

 activation_2 (Activation)   (None, 237, 237, 64)      0
['batch_normalization_2[0][0]'
                                                                  ]

 max_pooling2d (MaxPooling2  (None, 118, 118, 64)      0
['activation_2[0][0]']
 D)

 conv2d_3 (Conv2D)           (None, 118, 118, 80)      5120
['max_pooling2d[0][0]']

 batch_normalization_3 (Bat  (None, 118, 118, 80)      240
['conv2d_3[0][0]']
 chNormalization)
```

```
 activation_3 (Activation)   (None, 118, 118, 80)         0
['batch_normalization_3[0][0]'
                                                                  ]

 conv2d_4 (Conv2D)           (None, 116, 116, 192)        138240
['activation_3[0][0]']

 batch_normalization_4 (Bat  (None, 116, 116, 192)        576
['conv2d_4[0][0]']
 chNormalization)

 activation_4 (Activation)   (None, 116, 116, 192)        0
['batch_normalization_4[0][0]'
                                                                  ]

 max_pooling2d_1 (MaxPoolin  (None, 57, 57, 192)          0
['activation_4[0][0]']
 g2D)

 conv2d_8 (Conv2D)           (None, 57, 57, 64)           12288
['max_pooling2d_1[0][0]']

 batch_normalization_8 (Bat  (None, 57, 57, 64)           192
['conv2d_8[0][0]']
 chNormalization)

 activation_8 (Activation)   (None, 57, 57, 64)           0
['batch_normalization_8[0][0]'
                                                                  ]

 conv2d_6 (Conv2D)           (None, 57, 57, 48)           9216
['max_pooling2d_1[0][0]']

 conv2d_9 (Conv2D)           (None, 57, 57, 96)           55296
['activation_8[0][0]']

 batch_normalization_6 (Bat  (None, 57, 57, 48)           144
['conv2d_6[0][0]']
 chNormalization)

 batch_normalization_9 (Bat  (None, 57, 57, 96)           288
['conv2d_9[0][0]']
 chNormalization)

 activation_6 (Activation)   (None, 57, 57, 48)           0
['batch_normalization_6[0][0]'
                                                                  ]
```

```
 activation_9 (Activation)   (None, 57, 57, 96)          0
['batch_normalization_9[0][0]'
                                                                 ]

 average_pooling2d (Average  (None, 57, 57, 192)         0
['max_pooling2d_1[0][0]']
 Pooling2D)

 conv2d_5 (Conv2D)           (None, 57, 57, 64)          12288
['max_pooling2d_1[0][0]']

 conv2d_7 (Conv2D)           (None, 57, 57, 64)          76800
['activation_6[0][0]']

 conv2d_10 (Conv2D)          (None, 57, 57, 96)          82944
['activation_9[0][0]']

 conv2d_11 (Conv2D)          (None, 57, 57, 32)          6144
['average_pooling2d[0][0]']

 batch_normalization_5 (Bat  (None, 57, 57, 64)          192
['conv2d_5[0][0]']
 chNormalization)

 batch_normalization_7 (Bat  (None, 57, 57, 64)          192
['conv2d_7[0][0]']
 chNormalization)

 batch_normalization_10 (Ba  (None, 57, 57, 96)          288
['conv2d_10[0][0]']
 tchNormalization)

 batch_normalization_11 (Ba  (None, 57, 57, 32)          96
['conv2d_11[0][0]']
 tchNormalization)

 activation_5 (Activation)   (None, 57, 57, 64)          0
['batch_normalization_5[0][0]'
                                                                 ]

 activation_7 (Activation)   (None, 57, 57, 64)          0
['batch_normalization_7[0][0]'
                                                                 ]

 activation_10 (Activation)  (None, 57, 57, 96)          0
['batch_normalization_10[0][0]
                                                                 ']
```

```
 activation_11 (Activation)  (None, 57, 57, 32)           0
['batch_normalization_11[0][0]
                                                                 ']

 mixed0 (Concatenate)        (None, 57, 57, 256)          0
['activation_5[0][0]',
'activation_7[0][0]',
'activation_10[0][0]',
'activation_11[0][0]']

 conv2d_15 (Conv2D)          (None, 57, 57, 64)           16384
['mixed0[0][0]']

 batch_normalization_15 (Ba  (None, 57, 57, 64)           192
['conv2d_15[0][0]']
 tchNormalization)

 activation_15 (Activation)  (None, 57, 57, 64)           0
['batch_normalization_15[0][0]
                                                                 ']

 conv2d_13 (Conv2D)          (None, 57, 57, 48)           12288
['mixed0[0][0]']

 conv2d_16 (Conv2D)          (None, 57, 57, 96)           55296
['activation_15[0][0]']

 batch_normalization_13 (Ba  (None, 57, 57, 48)           144
['conv2d_13[0][0]']
 tchNormalization)

 batch_normalization_16 (Ba  (None, 57, 57, 96)           288
['conv2d_16[0][0]']
 tchNormalization)

 activation_13 (Activation)  (None, 57, 57, 48)           0
['batch_normalization_13[0][0]
                                                                 ']

 activation_16 (Activation)  (None, 57, 57, 96)           0
['batch_normalization_16[0][0]
                                                                 ']

 average_pooling2d_1 (Avera  (None, 57, 57, 256)          0
['mixed0[0][0]']
 gePooling2D)
```

```
 conv2d_12 (Conv2D)          (None, 57, 57, 64)          16384
['mixed0[0][0]']

 conv2d_14 (Conv2D)          (None, 57, 57, 64)          76800
['activation_13[0][0]']

 conv2d_17 (Conv2D)          (None, 57, 57, 96)          82944
['activation_16[0][0]']

 conv2d_18 (Conv2D)          (None, 57, 57, 64)          16384
['average_pooling2d_1[0][0]']

 batch_normalization_12 (Ba  (None, 57, 57, 64)          192
['conv2d_12[0][0]']
 tchNormalization)

 batch_normalization_14 (Ba  (None, 57, 57, 64)          192
['conv2d_14[0][0]']
 tchNormalization)

 batch_normalization_17 (Ba  (None, 57, 57, 96)          288
['conv2d_17[0][0]']
 tchNormalization)

 batch_normalization_18 (Ba  (None, 57, 57, 64)          192
['conv2d_18[0][0]']
 tchNormalization)

 activation_12 (Activation)  (None, 57, 57, 64)          0
['batch_normalization_12[0][0]
                                                                  ']

 activation_14 (Activation)  (None, 57, 57, 64)          0
['batch_normalization_14[0][0]
                                                                  ']

 activation_17 (Activation)  (None, 57, 57, 96)          0
['batch_normalization_17[0][0]
                                                                  ']

 activation_18 (Activation)  (None, 57, 57, 64)          0
['batch_normalization_18[0][0]
                                                                  ']

 mixed1 (Concatenate)        (None, 57, 57, 288)         0
['activation_12[0][0]',
'activation_14[0][0]',
'activation_17[0][0]',
```

```
'activation_18[0][0]']

 conv2d_22 (Conv2D)          (None, 57, 57, 64)           18432
['mixed1[0][0]']

 batch_normalization_22 (Ba  (None, 57, 57, 64)           192
['conv2d_22[0][0]']
 tchNormalization)

 activation_22 (Activation)  (None, 57, 57, 64)           0
['batch_normalization_22[0][0]
                                                                  ']

 conv2d_20 (Conv2D)          (None, 57, 57, 48)           13824
['mixed1[0][0]']

 conv2d_23 (Conv2D)          (None, 57, 57, 96)           55296
['activation_22[0][0]']

 batch_normalization_20 (Ba  (None, 57, 57, 48)           144
['conv2d_20[0][0]']
 tchNormalization)

 batch_normalization_23 (Ba  (None, 57, 57, 96)           288
['conv2d_23[0][0]']
 tchNormalization)

 activation_20 (Activation)  (None, 57, 57, 48)           0
['batch_normalization_20[0][0]
                                                                  ']

 activation_23 (Activation)  (None, 57, 57, 96)           0
['batch_normalization_23[0][0]
                                                                  ']

 average_pooling2d_2 (Avera  (None, 57, 57, 288)          0
['mixed1[0][0]']
 gePooling2D)

 conv2d_19 (Conv2D)          (None, 57, 57, 64)           18432
['mixed1[0][0]']

 conv2d_21 (Conv2D)          (None, 57, 57, 64)           76800
['activation_20[0][0]']

 conv2d_24 (Conv2D)          (None, 57, 57, 96)           82944
['activation_23[0][0]']
```

```
 conv2d_25 (Conv2D)          (None, 57, 57, 64)          18432
['average_pooling2d_2[0][0]']

 batch_normalization_19 (Ba  (None, 57, 57, 64)          192
['conv2d_19[0][0]']
 tchNormalization)

 batch_normalization_21 (Ba  (None, 57, 57, 64)          192
['conv2d_21[0][0]']
 tchNormalization)

 batch_normalization_24 (Ba  (None, 57, 57, 96)          288
['conv2d_24[0][0]']
 tchNormalization)

 batch_normalization_25 (Ba  (None, 57, 57, 64)          192
['conv2d_25[0][0]']
 tchNormalization)

 activation_19 (Activation)  (None, 57, 57, 64)          0
['batch_normalization_19[0][0]
                                                                  ']

 activation_21 (Activation)  (None, 57, 57, 64)          0
['batch_normalization_21[0][0]
                                                                  ']

 activation_24 (Activation)  (None, 57, 57, 96)          0
['batch_normalization_24[0][0]
                                                                  ']

 activation_25 (Activation)  (None, 57, 57, 64)          0
['batch_normalization_25[0][0]
                                                                  ']

 mixed2 (Concatenate)        (None, 57, 57, 288)         0
['activation_19[0][0]',
'activation_21[0][0]',
'activation_24[0][0]',
'activation_25[0][0]']

 conv2d_27 (Conv2D)          (None, 57, 57, 64)          18432
['mixed2[0][0]']

 batch_normalization_27 (Ba  (None, 57, 57, 64)          192
['conv2d_27[0][0]']
 tchNormalization)
```

```
 activation_27 (Activation)  (None, 57, 57, 64)           0
['batch_normalization_27[0][0]
                                                                 ']

 conv2d_28 (Conv2D)          (None, 57, 57, 96)           55296
['activation_27[0][0]']

 batch_normalization_28 (Ba  (None, 57, 57, 96)           288
['conv2d_28[0][0]']
 tchNormalization)

 activation_28 (Activation)  (None, 57, 57, 96)           0
['batch_normalization_28[0][0]
                                                                 ']

 conv2d_26 (Conv2D)          (None, 28, 28, 384)          995328
['mixed2[0][0]']

 conv2d_29 (Conv2D)          (None, 28, 28, 96)           82944
['activation_28[0][0]']

 batch_normalization_26 (Ba  (None, 28, 28, 384)          1152
['conv2d_26[0][0]']
 tchNormalization)

 batch_normalization_29 (Ba  (None, 28, 28, 96)           288
['conv2d_29[0][0]']
 tchNormalization)

 activation_26 (Activation)  (None, 28, 28, 384)          0
['batch_normalization_26[0][0]
                                                                 ']

 activation_29 (Activation)  (None, 28, 28, 96)           0
['batch_normalization_29[0][0]
                                                                 ']

 max_pooling2d_2 (MaxPoolin  (None, 28, 28, 288)          0
['mixed2[0][0]']
 g2D)

 mixed3 (Concatenate)        (None, 28, 28, 768)          0
['activation_26[0][0]',
'activation_29[0][0]',
'max_pooling2d_2[0][0]']

 conv2d_34 (Conv2D)          (None, 28, 28, 128)          98304
['mixed3[0][0]']
```

```
 batch_normalization_34 (Ba  (None, 28, 28, 128)          384
['conv2d_34[0][0]']
 tchNormalization)

 activation_34 (Activation)  (None, 28, 28, 128)          0
['batch_normalization_34[0][0]
                                                                  ']

 conv2d_35 (Conv2D)          (None, 28, 28, 128)          114688
['activation_34[0][0]']

 batch_normalization_35 (Ba  (None, 28, 28, 128)          384
['conv2d_35[0][0]']
 tchNormalization)

 activation_35 (Activation)  (None, 28, 28, 128)          0
['batch_normalization_35[0][0]
                                                                  ']

 conv2d_31 (Conv2D)          (None, 28, 28, 128)          98304
['mixed3[0][0]']

 conv2d_36 (Conv2D)          (None, 28, 28, 128)          114688
['activation_35[0][0]']

 batch_normalization_31 (Ba  (None, 28, 28, 128)          384
['conv2d_31[0][0]']
 tchNormalization)

 batch_normalization_36 (Ba  (None, 28, 28, 128)          384
['conv2d_36[0][0]']
 tchNormalization)

 activation_31 (Activation)  (None, 28, 28, 128)          0
['batch_normalization_31[0][0]
                                                                  ']

 activation_36 (Activation)  (None, 28, 28, 128)          0
['batch_normalization_36[0][0]
                                                                  ']

 conv2d_32 (Conv2D)          (None, 28, 28, 128)          114688
['activation_31[0][0]']

 conv2d_37 (Conv2D)          (None, 28, 28, 128)          114688
['activation_36[0][0]']
```

```
 batch_normalization_32 (Ba  (None, 28, 28, 128)          384
['conv2d_32[0][0]']
 tchNormalization)

 batch_normalization_37 (Ba  (None, 28, 28, 128)          384
['conv2d_37[0][0]']
 tchNormalization)

 activation_32 (Activation)  (None, 28, 28, 128)          0
['batch_normalization_32[0][0]
                                                                  ']

 activation_37 (Activation)  (None, 28, 28, 128)          0
['batch_normalization_37[0][0]
                                                                  ']

 average_pooling2d_3 (Avera  (None, 28, 28, 768)          0
['mixed3[0][0]']
 gePooling2D)

 conv2d_30 (Conv2D)          (None, 28, 28, 192)          147456
['mixed3[0][0]']

 conv2d_33 (Conv2D)          (None, 28, 28, 192)          172032
['activation_32[0][0]']

 conv2d_38 (Conv2D)          (None, 28, 28, 192)          172032
['activation_37[0][0]']

 conv2d_39 (Conv2D)          (None, 28, 28, 192)          147456
['average_pooling2d_3[0][0]']

 batch_normalization_30 (Ba  (None, 28, 28, 192)          576
['conv2d_30[0][0]']
 tchNormalization)

 batch_normalization_33 (Ba  (None, 28, 28, 192)          576
['conv2d_33[0][0]']
 tchNormalization)

 batch_normalization_38 (Ba  (None, 28, 28, 192)          576
['conv2d_38[0][0]']
 tchNormalization)

 batch_normalization_39 (Ba  (None, 28, 28, 192)          576
['conv2d_39[0][0]']
 tchNormalization)
```

```
 activation_30 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_30[0][0]
                                                                 ']

 activation_33 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_33[0][0]
                                                                 ']

 activation_38 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_38[0][0]
                                                                 ']

 activation_39 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_39[0][0]
                                                                 ']

 mixed4 (Concatenate)        (None, 28, 28, 768)          0
['activation_30[0][0]',
'activation_33[0][0]',
'activation_38[0][0]',
'activation_39[0][0]']

 conv2d_44 (Conv2D)          (None, 28, 28, 160)          122880
['mixed4[0][0]']

 batch_normalization_44 (Ba  (None, 28, 28, 160)          480
['conv2d_44[0][0]']
 tchNormalization)

 activation_44 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_44[0][0]
                                                                 ']

 conv2d_45 (Conv2D)          (None, 28, 28, 160)          179200
['activation_44[0][0]']

 batch_normalization_45 (Ba  (None, 28, 28, 160)          480
['conv2d_45[0][0]']
 tchNormalization)

 activation_45 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_45[0][0]
                                                                 ']

 conv2d_41 (Conv2D)          (None, 28, 28, 160)          122880
['mixed4[0][0]']

 conv2d_46 (Conv2D)          (None, 28, 28, 160)          179200
```

```
['activation_45[0][0]']

 batch_normalization_41 (Ba  (None, 28, 28, 160)          480
['conv2d_41[0][0]']
 tchNormalization)

 batch_normalization_46 (Ba  (None, 28, 28, 160)          480
['conv2d_46[0][0]']
 tchNormalization)

 activation_41 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_41[0][0]
                                                                  ']

 activation_46 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_46[0][0]
                                                                  ']

 conv2d_42 (Conv2D)          (None, 28, 28, 160)          179200
['activation_41[0][0]']

 conv2d_47 (Conv2D)          (None, 28, 28, 160)          179200
['activation_46[0][0]']

 batch_normalization_42 (Ba  (None, 28, 28, 160)          480
['conv2d_42[0][0]']
 tchNormalization)

 batch_normalization_47 (Ba  (None, 28, 28, 160)          480
['conv2d_47[0][0]']
 tchNormalization)

 activation_42 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_42[0][0]
                                                                  ']

 activation_47 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_47[0][0]
                                                                  ']

 average_pooling2d_4 (Avera  (None, 28, 28, 768)          0
['mixed4[0][0]']
 gePooling2D)

 conv2d_40 (Conv2D)          (None, 28, 28, 192)          147456
['mixed4[0][0]']

 conv2d_43 (Conv2D)          (None, 28, 28, 192)          215040
```

```
['activation_42[0][0]']

 conv2d_48 (Conv2D)          (None, 28, 28, 192)          215040
['activation_47[0][0]']

 conv2d_49 (Conv2D)          (None, 28, 28, 192)          147456
['average_pooling2d_4[0][0]']

 batch_normalization_40 (Ba  (None, 28, 28, 192)          576
['conv2d_40[0][0]']
 tchNormalization)

 batch_normalization_43 (Ba  (None, 28, 28, 192)          576
['conv2d_43[0][0]']
 tchNormalization)

 batch_normalization_48 (Ba  (None, 28, 28, 192)          576
['conv2d_48[0][0]']
 tchNormalization)

 batch_normalization_49 (Ba  (None, 28, 28, 192)          576
['conv2d_49[0][0]']
 tchNormalization)

 activation_40 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_40[0][0]
                                                                  ']

 activation_43 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_43[0][0]
                                                                  ']

 activation_48 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_48[0][0]
                                                                  ']

 activation_49 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_49[0][0]
                                                                  ']

 mixed5 (Concatenate)        (None, 28, 28, 768)          0
['activation_40[0][0]',
'activation_43[0][0]',
'activation_48[0][0]',
'activation_49[0][0]']

 conv2d_54 (Conv2D)          (None, 28, 28, 160)          122880
['mixed5[0][0]']
```

```
 batch_normalization_54 (Ba  (None, 28, 28, 160)          480
['conv2d_54[0][0]']
 tchNormalization)

 activation_54 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_54[0][0]
                                                                  ']

 conv2d_55 (Conv2D)          (None, 28, 28, 160)          179200
['activation_54[0][0]']

 batch_normalization_55 (Ba  (None, 28, 28, 160)          480
['conv2d_55[0][0]']
 tchNormalization)

 activation_55 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_55[0][0]
                                                                  ']

 conv2d_51 (Conv2D)          (None, 28, 28, 160)          122880
['mixed5[0][0]']

 conv2d_56 (Conv2D)          (None, 28, 28, 160)          179200
['activation_55[0][0]']

 batch_normalization_51 (Ba  (None, 28, 28, 160)          480
['conv2d_51[0][0]']
 tchNormalization)

 batch_normalization_56 (Ba  (None, 28, 28, 160)          480
['conv2d_56[0][0]']
 tchNormalization)

 activation_51 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_51[0][0]
                                                                  ']

 activation_56 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_56[0][0]
                                                                  ']

 conv2d_52 (Conv2D)          (None, 28, 28, 160)          179200
['activation_51[0][0]']

 conv2d_57 (Conv2D)          (None, 28, 28, 160)          179200
['activation_56[0][0]']
```

```
 batch_normalization_52 (Ba  (None, 28, 28, 160)          480
['conv2d_52[0][0]']
 tchNormalization)

 batch_normalization_57 (Ba  (None, 28, 28, 160)          480
['conv2d_57[0][0]']
 tchNormalization)

 activation_52 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_52[0][0]
                                                                  ']

 activation_57 (Activation)  (None, 28, 28, 160)          0
['batch_normalization_57[0][0]
                                                                  ']

 average_pooling2d_5 (Avera  (None, 28, 28, 768)          0
['mixed5[0][0]']
 gePooling2D)

 conv2d_50 (Conv2D)          (None, 28, 28, 192)          147456
['mixed5[0][0]']

 conv2d_53 (Conv2D)          (None, 28, 28, 192)          215040
['activation_52[0][0]']

 conv2d_58 (Conv2D)          (None, 28, 28, 192)          215040
['activation_57[0][0]']

 conv2d_59 (Conv2D)          (None, 28, 28, 192)          147456
['average_pooling2d_5[0][0]']

 batch_normalization_50 (Ba  (None, 28, 28, 192)          576
['conv2d_50[0][0]']
 tchNormalization)

 batch_normalization_53 (Ba  (None, 28, 28, 192)          576
['conv2d_53[0][0]']
 tchNormalization)

 batch_normalization_58 (Ba  (None, 28, 28, 192)          576
['conv2d_58[0][0]']
 tchNormalization)

 batch_normalization_59 (Ba  (None, 28, 28, 192)          576
['conv2d_59[0][0]']
 tchNormalization)
```

```
 activation_50 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_50[0][0]
                                                                 ']

 activation_53 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_53[0][0]
                                                                 ']

 activation_58 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_58[0][0]
                                                                 ']

 activation_59 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_59[0][0]
                                                                 ']

 mixed6 (Concatenate)        (None, 28, 28, 768)          0
['activation_50[0][0]',
'activation_53[0][0]',
'activation_58[0][0]',
'activation_59[0][0]']

 conv2d_64 (Conv2D)          (None, 28, 28, 192)          147456
['mixed6[0][0]']

 batch_normalization_64 (Ba  (None, 28, 28, 192)          576
['conv2d_64[0][0]']
 tchNormalization)

 activation_64 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_64[0][0]
                                                                 ']

 conv2d_65 (Conv2D)          (None, 28, 28, 192)          258048
['activation_64[0][0]']

 batch_normalization_65 (Ba  (None, 28, 28, 192)          576
['conv2d_65[0][0]']
 tchNormalization)

 activation_65 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_65[0][0]
                                                                 ']

 conv2d_61 (Conv2D)          (None, 28, 28, 192)          147456
['mixed6[0][0]']

 conv2d_66 (Conv2D)          (None, 28, 28, 192)          258048
```

```
['activation_65[0][0]']

 batch_normalization_61 (Ba  (None, 28, 28, 192)          576
['conv2d_61[0][0]']
 tchNormalization)

 batch_normalization_66 (Ba  (None, 28, 28, 192)          576
['conv2d_66[0][0]']
 tchNormalization)

 activation_61 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_61[0][0]
                                                                  ']

 activation_66 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_66[0][0]
                                                                  ']

 conv2d_62 (Conv2D)          (None, 28, 28, 192)          258048
['activation_61[0][0]']

 conv2d_67 (Conv2D)          (None, 28, 28, 192)          258048
['activation_66[0][0]']

 batch_normalization_62 (Ba  (None, 28, 28, 192)          576
['conv2d_62[0][0]']
 tchNormalization)

 batch_normalization_67 (Ba  (None, 28, 28, 192)          576
['conv2d_67[0][0]']
 tchNormalization)

 activation_62 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_62[0][0]
                                                                  ']

 activation_67 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_67[0][0]
                                                                  ']

 average_pooling2d_6 (Avera  (None, 28, 28, 768)          0
['mixed6[0][0]']
 gePooling2D)

 conv2d_60 (Conv2D)          (None, 28, 28, 192)          147456
['mixed6[0][0]']

 conv2d_63 (Conv2D)          (None, 28, 28, 192)          258048
```

```
['activation_62[0][0]']

 conv2d_68 (Conv2D)          (None, 28, 28, 192)          258048
['activation_67[0][0]']

 conv2d_69 (Conv2D)          (None, 28, 28, 192)          147456
['average_pooling2d_6[0][0]']

 batch_normalization_60 (Ba  (None, 28, 28, 192)          576
['conv2d_60[0][0]']
 tchNormalization)

 batch_normalization_63 (Ba  (None, 28, 28, 192)          576
['conv2d_63[0][0]']
 tchNormalization)

 batch_normalization_68 (Ba  (None, 28, 28, 192)          576
['conv2d_68[0][0]']
 tchNormalization)

 batch_normalization_69 (Ba  (None, 28, 28, 192)          576
['conv2d_69[0][0]']
 tchNormalization)

 activation_60 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_60[0][0]
                                                                  ']

 activation_63 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_63[0][0]
                                                                  ']

 activation_68 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_68[0][0]
                                                                  ']

 activation_69 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_69[0][0]
                                                                  ']

 mixed7 (Concatenate)        (None, 28, 28, 768)          0
['activation_60[0][0]',
'activation_63[0][0]',
'activation_68[0][0]',
'activation_69[0][0]']

 conv2d_72 (Conv2D)          (None, 28, 28, 192)          147456
['mixed7[0][0]']
```

```
 batch_normalization_72 (Ba  (None, 28, 28, 192)          576
['conv2d_72[0][0]']
 tchNormalization)

 activation_72 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_72[0][0]
                                                                  ']

 conv2d_73 (Conv2D)          (None, 28, 28, 192)          258048
['activation_72[0][0]']

 batch_normalization_73 (Ba  (None, 28, 28, 192)          576
['conv2d_73[0][0]']
 tchNormalization)

 activation_73 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_73[0][0]
                                                                  ']

 conv2d_70 (Conv2D)          (None, 28, 28, 192)          147456
['mixed7[0][0]']

 conv2d_74 (Conv2D)          (None, 28, 28, 192)          258048
['activation_73[0][0]']

 batch_normalization_70 (Ba  (None, 28, 28, 192)          576
['conv2d_70[0][0]']
 tchNormalization)

 batch_normalization_74 (Ba  (None, 28, 28, 192)          576
['conv2d_74[0][0]']
 tchNormalization)

 activation_70 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_70[0][0]
                                                                  ']

 activation_74 (Activation)  (None, 28, 28, 192)          0
['batch_normalization_74[0][0]
                                                                  ']

 conv2d_71 (Conv2D)          (None, 13, 13, 320)          552960
['activation_70[0][0]']

 conv2d_75 (Conv2D)          (None, 13, 13, 192)          331776
['activation_74[0][0]']
```

```
 batch_normalization_71 (Ba  (None, 13, 13, 320)          960
['conv2d_71[0][0]']
 tchNormalization)

 batch_normalization_75 (Ba  (None, 13, 13, 192)          576
['conv2d_75[0][0]']
 tchNormalization)

 activation_71 (Activation)  (None, 13, 13, 320)          0
['batch_normalization_71[0][0]
                                                                  ']

 activation_75 (Activation)  (None, 13, 13, 192)          0
['batch_normalization_75[0][0]
                                                                  ']

 max_pooling2d_3 (MaxPoolin  (None, 13, 13, 768)          0
['mixed7[0][0]']
 g2D)

 mixed8 (Concatenate)        (None, 13, 13, 1280)         0
['activation_71[0][0]',
'activation_75[0][0]',
'max_pooling2d_3[0][0]']

 conv2d_80 (Conv2D)          (None, 13, 13, 448)          573440
['mixed8[0][0]']

 batch_normalization_80 (Ba  (None, 13, 13, 448)          1344
['conv2d_80[0][0]']
 tchNormalization)

 activation_80 (Activation)  (None, 13, 13, 448)          0
['batch_normalization_80[0][0]
                                                                  ']

 conv2d_77 (Conv2D)          (None, 13, 13, 384)          491520
['mixed8[0][0]']

 conv2d_81 (Conv2D)          (None, 13, 13, 384)          1548288
['activation_80[0][0]']

 batch_normalization_77 (Ba  (None, 13, 13, 384)          1152
['conv2d_77[0][0]']
 tchNormalization)

 batch_normalization_81 (Ba  (None, 13, 13, 384)          1152
['conv2d_81[0][0]']
```

```
 tchNormalization)

 activation_77 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_77[0][0]
                                                                 ']

 activation_81 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_81[0][0]
                                                                 ']

 conv2d_78 (Conv2D)          (None, 13, 13, 384)          442368
['activation_77[0][0]']

 conv2d_79 (Conv2D)          (None, 13, 13, 384)          442368
['activation_77[0][0]']

 conv2d_82 (Conv2D)          (None, 13, 13, 384)          442368
['activation_81[0][0]']

 conv2d_83 (Conv2D)          (None, 13, 13, 384)          442368
['activation_81[0][0]']

 average_pooling2d_7 (Avera  (None, 13, 13, 1280)         0
['mixed8[0][0]']
 gePooling2D)

 conv2d_76 (Conv2D)          (None, 13, 13, 320)          409600
['mixed8[0][0]']

 batch_normalization_78 (Ba  (None, 13, 13, 384)          1152
['conv2d_78[0][0]']
 tchNormalization)

 batch_normalization_79 (Ba  (None, 13, 13, 384)          1152
['conv2d_79[0][0]']
 tchNormalization)

 batch_normalization_82 (Ba  (None, 13, 13, 384)          1152
['conv2d_82[0][0]']
 tchNormalization)

 batch_normalization_83 (Ba  (None, 13, 13, 384)          1152
['conv2d_83[0][0]']
 tchNormalization)

 conv2d_84 (Conv2D)          (None, 13, 13, 192)          245760
['average_pooling2d_7[0][0]']
```

```
 batch_normalization_76 (Ba  (None, 13, 13, 320)          960
['conv2d_76[0][0]']
 tchNormalization)

 activation_78 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_78[0][0]
                                                                  ']

 activation_79 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_79[0][0]
                                                                  ']

 activation_82 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_82[0][0]
                                                                  ']

 activation_83 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_83[0][0]
                                                                  ']

 batch_normalization_84 (Ba  (None, 13, 13, 192)          576
['conv2d_84[0][0]']
 tchNormalization)

 activation_76 (Activation)  (None, 13, 13, 320)          0
['batch_normalization_76[0][0]
                                                                  ']

 mixed9_0 (Concatenate)      (None, 13, 13, 768)          0
['activation_78[0][0]',
'activation_79[0][0]']

 concatenate (Concatenate)   (None, 13, 13, 768)          0
['activation_82[0][0]',
'activation_83[0][0]']

 activation_84 (Activation)  (None, 13, 13, 192)          0
['batch_normalization_84[0][0]
                                                                  ']

 mixed9 (Concatenate)        (None, 13, 13, 2048)         0
['activation_76[0][0]',
'mixed9_0[0][0]',
'concatenate[0][0]',
'activation_84[0][0]']

 conv2d_89 (Conv2D)          (None, 13, 13, 448)          917504
['mixed9[0][0]']
```

```
 batch_normalization_89 (Ba  (None, 13, 13, 448)          1344
['conv2d_89[0][0]']
 tchNormalization)

 activation_89 (Activation)  (None, 13, 13, 448)          0
['batch_normalization_89[0][0]
                                                                  ']

 conv2d_86 (Conv2D)          (None, 13, 13, 384)          786432
['mixed9[0][0]']

 conv2d_90 (Conv2D)          (None, 13, 13, 384)          1548288
['activation_89[0][0]']

 batch_normalization_86 (Ba  (None, 13, 13, 384)          1152
['conv2d_86[0][0]']
 tchNormalization)

 batch_normalization_90 (Ba  (None, 13, 13, 384)          1152
['conv2d_90[0][0]']
 tchNormalization)

 activation_86 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_86[0][0]
                                                                  ']

 activation_90 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_90[0][0]
                                                                  ']

 conv2d_87 (Conv2D)          (None, 13, 13, 384)          442368
['activation_86[0][0]']

 conv2d_88 (Conv2D)          (None, 13, 13, 384)          442368
['activation_86[0][0]']

 conv2d_91 (Conv2D)          (None, 13, 13, 384)          442368
['activation_90[0][0]']

 conv2d_92 (Conv2D)          (None, 13, 13, 384)          442368
['activation_90[0][0]']

 average_pooling2d_8 (Avera  (None, 13, 13, 2048)         0
['mixed9[0][0]']
 gePooling2D)

 conv2d_85 (Conv2D)          (None, 13, 13, 320)          655360
```

```
['mixed9[0][0]']

 batch_normalization_87 (Ba  (None, 13, 13, 384)          1152
['conv2d_87[0][0]']
 tchNormalization)

 batch_normalization_88 (Ba  (None, 13, 13, 384)          1152
['conv2d_88[0][0]']
 tchNormalization)

 batch_normalization_91 (Ba  (None, 13, 13, 384)          1152
['conv2d_91[0][0]']
 tchNormalization)

 batch_normalization_92 (Ba  (None, 13, 13, 384)          1152
['conv2d_92[0][0]']
 tchNormalization)

 conv2d_93 (Conv2D)          (None, 13, 13, 192)          393216
['average_pooling2d_8[0][0]']

 batch_normalization_85 (Ba  (None, 13, 13, 320)          960
['conv2d_85[0][0]']
 tchNormalization)

 activation_87 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_87[0][0]
                                                                  ']

 activation_88 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_88[0][0]
                                                                  ']

 activation_91 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_91[0][0]
                                                                  ']

 activation_92 (Activation)  (None, 13, 13, 384)          0
['batch_normalization_92[0][0]
                                                                  ']

 batch_normalization_93 (Ba  (None, 13, 13, 192)          576
['conv2d_93[0][0]']
 tchNormalization)

 activation_85 (Activation)  (None, 13, 13, 320)          0
['batch_normalization_85[0][0]
                                                                  ']
```

```
 mixed9_1 (Concatenate)      (None, 13, 13, 768)          0
['activation_87[0][0]',
'activation_88[0][0]']

 concatenate_1 (Concatenate  (None, 13, 13, 768)          0
['activation_91[0][0]',
 )
'activation_92[0][0]']

 activation_93 (Activation)  (None, 13, 13, 192)          0
['batch_normalization_93[0][0]
                                                                  ']

 mixed10 (Concatenate)       (None, 13, 13, 2048)         0
['activation_85[0][0]',
'mixed9_1[0][0]',
'concatenate_1[0][0]',
'activation_93[0][0]']

 flatten (Flatten)           (None, 346112)               0
['mixed10[0][0]']

 dense (Dense)               (None, 256)                  8860492
['flatten[0][0]']
                                                         8

 dense_1 (Dense)             (None, 9)                    2313
['dense[0][0]']

==================================================================================
==================
Total params: 110410025 (421.18 MB)
Trainable params: 88607241 (338.01 MB)
Non-trainable params: 21802784 (83.17 MB)
----------------------------------------------------------------------------------
------------------
```

0.12 Early Stopping Callback Configuration

[19]:
```
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3,␣
 ↪min_lr=0.0001)
early_stop = EarlyStopping(monitor='val_loss', patience=5,␣
 ↪restore_best_weights=True)
```

0.13 Model Training

[20]:
```
model_history = model.fit(
    training_set,
    validation_data=val_set,
    epochs=20,
    steps_per_epoch=len(training_set),
    validation_steps=len(val_set),
    callbacks=[reduce_lr, early_stop])
```

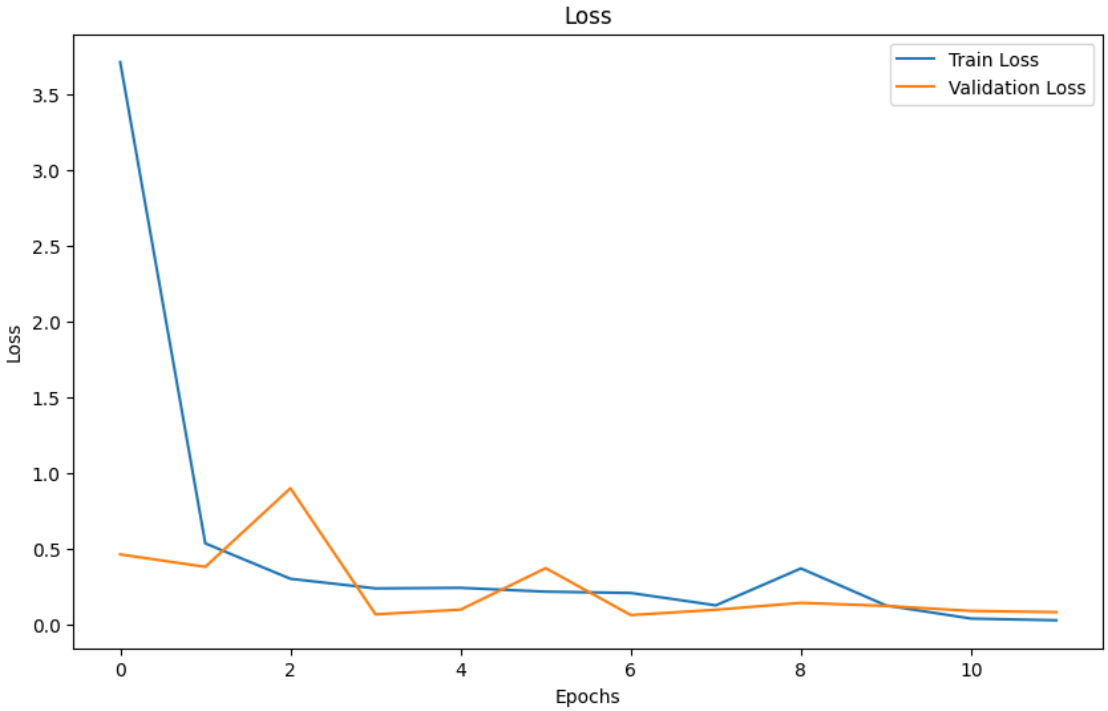
```
Epoch 1/20
271/271 [==============================] - 646s 2s/step - loss: 3.7135 -
accuracy: 0.8986 - precision: 0.9022 - recall: 0.8984 - f1_score: 0.8987 -
val_loss: 0.4623 - val_accuracy: 0.9772 - val_precision: 0.9772 - val_recall:
0.9772 - val_f1_score: 0.9773 - lr: 0.0010
Epoch 2/20
271/271 [==============================] - 281s 1s/step - loss: 0.5353 -
accuracy: 0.9651 - precision: 0.9651 - recall: 0.9651 - f1_score: 0.9651 -
val_loss: 0.3801 - val_accuracy: 0.9563 - val_precision: 0.9563 - val_recall:
0.9563 - val_f1_score: 0.9562 - lr: 0.0010
Epoch 3/20
271/271 [==============================] - 272s 1s/step - loss: 0.3014 -
accuracy: 0.9725 - precision: 0.9725 - recall: 0.9725 - f1_score: 0.9725 -
val_loss: 0.8994 - val_accuracy: 0.9474 - val_precision: 0.9474 - val_recall:
0.9464 - val_f1_score: 0.9467 - lr: 0.0010
Epoch 4/20
271/271 [==============================] - 270s 996ms/step - loss: 0.2379 -
accuracy: 0.9825 - precision: 0.9824 - recall: 0.9822 - f1_score: 0.9824 -
val_loss: 0.0667 - val_accuracy: 0.9960 - val_precision: 0.9960 - val_recall:
0.9960 - val_f1_score: 0.9960 - lr: 0.0010
Epoch 5/20
271/271 [==============================] - 274s 1s/step - loss: 0.2415 -
accuracy: 0.9820 - precision: 0.9822 - recall: 0.9820 - f1_score: 0.9820 -
val_loss: 0.0976 - val_accuracy: 0.9901 - val_precision: 0.9901 - val_recall:
0.9901 - val_f1_score: 0.9901 - lr: 0.0010
Epoch 6/20
271/271 [==============================] - 267s 986ms/step - loss: 0.2164 -
accuracy: 0.9815 - precision: 0.9818 - recall: 0.9815 - f1_score: 0.9815 -
val_loss: 0.3715 - val_accuracy: 0.9732 - val_precision: 0.9732 - val_recall:
0.9732 - val_f1_score: 0.9737 - lr: 0.0010
Epoch 7/20
271/271 [==============================] - 272s 1s/step - loss: 0.2076 -
accuracy: 0.9827 - precision: 0.9827 - recall: 0.9827 - f1_score: 0.9827 -
val_loss: 0.0617 - val_accuracy: 0.9970 - val_precision: 0.9970 - val_recall:
0.9970 - val_f1_score: 0.9970 - lr: 0.0010
Epoch 8/20
271/271 [==============================] - 275s 1s/step - loss: 0.1261 -
accuracy: 0.9889 - precision: 0.9889 - recall: 0.9889 - f1_score: 0.9889 -
```

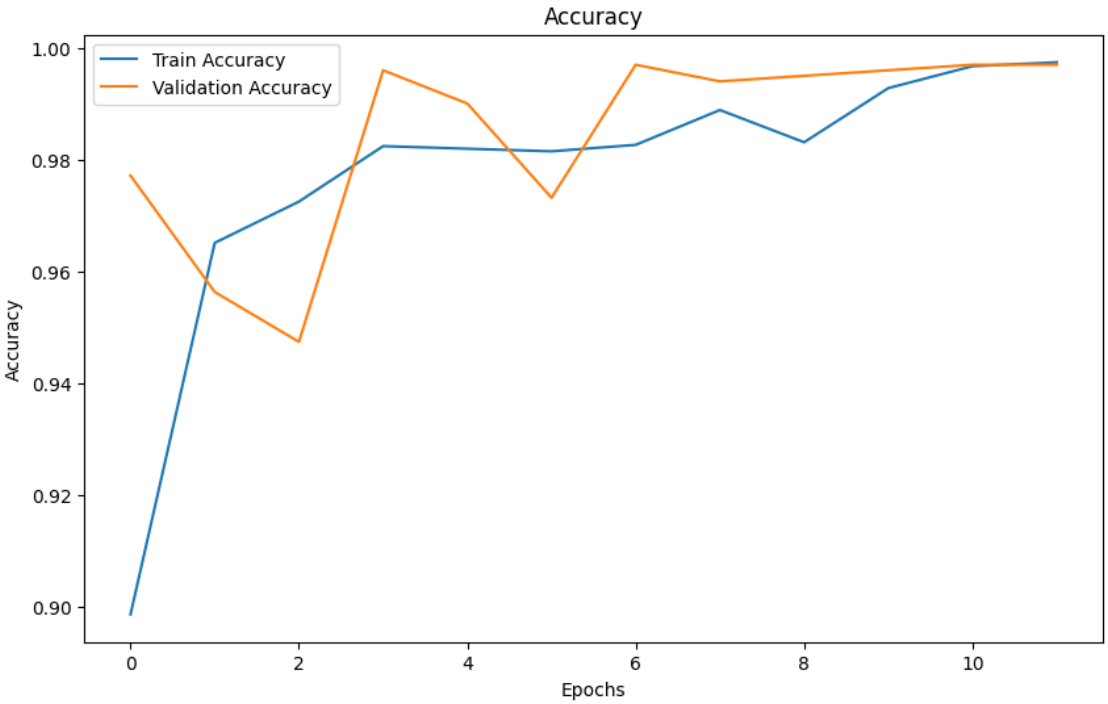
```
val_loss: 0.0967 - val_accuracy: 0.9940 - val_precision: 0.9940 - val_recall:
0.9940 - val_f1_score: 0.9940 - lr: 0.0010
Epoch 9/20
271/271 [==============================] - 272s 1s/step - loss: 0.3697 -
accuracy: 0.9831 - precision: 0.9831 - recall: 0.9831 - f1_score: 0.9831 -
val_loss: 0.1425 - val_accuracy: 0.9950 - val_precision: 0.9950 - val_recall:
0.9950 - val_f1_score: 0.9950 - lr: 0.0010
Epoch 10/20
271/271 [==============================] - 270s 998ms/step - loss: 0.1251 -
accuracy: 0.9928 - precision: 0.9928 - recall: 0.9928 - f1_score: 0.9928 -
val_loss: 0.1222 - val_accuracy: 0.9960 - val_precision: 0.9960 - val_recall:
0.9960 - val_f1_score: 0.9960 - lr: 0.0010
Epoch 11/20
271/271 [==============================] - 274s 1s/step - loss: 0.0386 -
accuracy: 0.9968 - precision: 0.9968 - recall: 0.9968 - f1_score: 0.9968 -
val_loss: 0.0892 - val_accuracy: 0.9970 - val_precision: 0.9970 - val_recall:
0.9970 - val_f1_score: 0.9970 - lr: 2.0000e-04
Epoch 12/20
271/271 [==============================] - 278s 1s/step - loss: 0.0273 -
accuracy: 0.9975 - precision: 0.9975 - recall: 0.9975 - f1_score: 0.9975 -
val_loss: 0.0811 - val_accuracy: 0.9970 - val_precision: 0.9970 - val_recall:
0.9970 - val_f1_score: 0.9970 - lr: 2.0000e-04
```

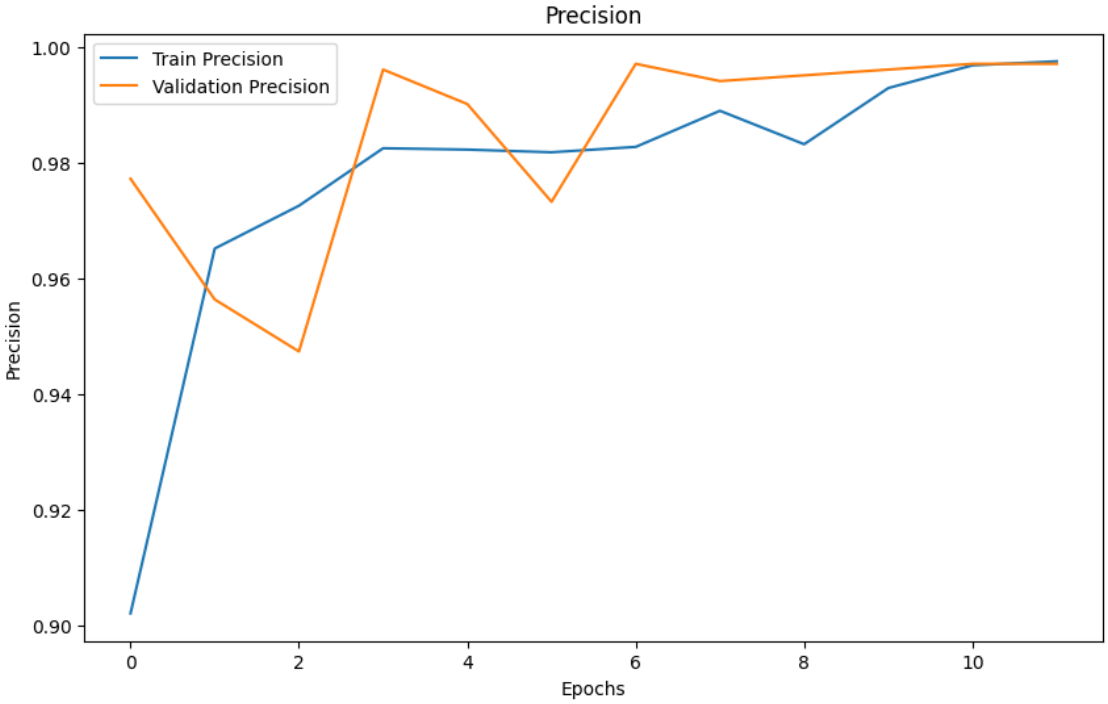
0.14 Model Evaluation Plots

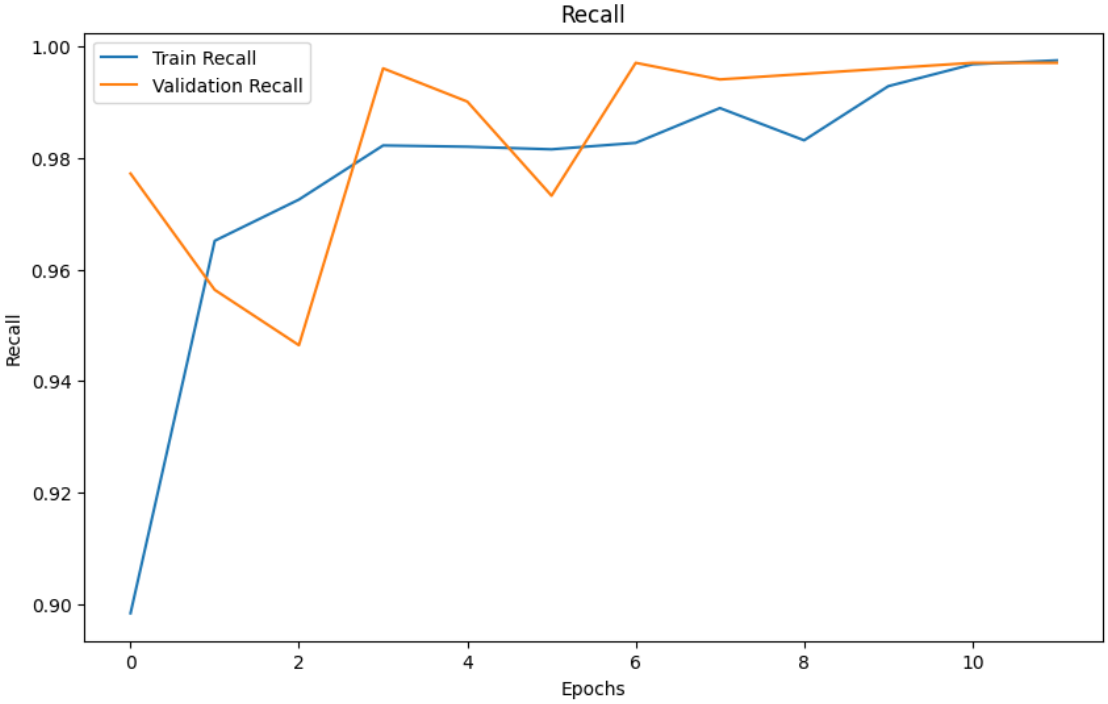
[21]:
```python
# Loss
plt.figure(figsize=(10, 6))
plt.plot(model_history.history['loss'], label='Train Loss')
plt.plot(model_history.history['val_loss'], label='Validation Loss')
plt.title('Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.savefig('LossVal_loss.png')
plt.show()

# Accuracy
plt.figure(figsize=(10, 6))
plt.plot(model_history.history['accuracy'], label='Train Accuracy')
plt.plot(model_history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.savefig('AccVal_acc.png')
plt.show()
```

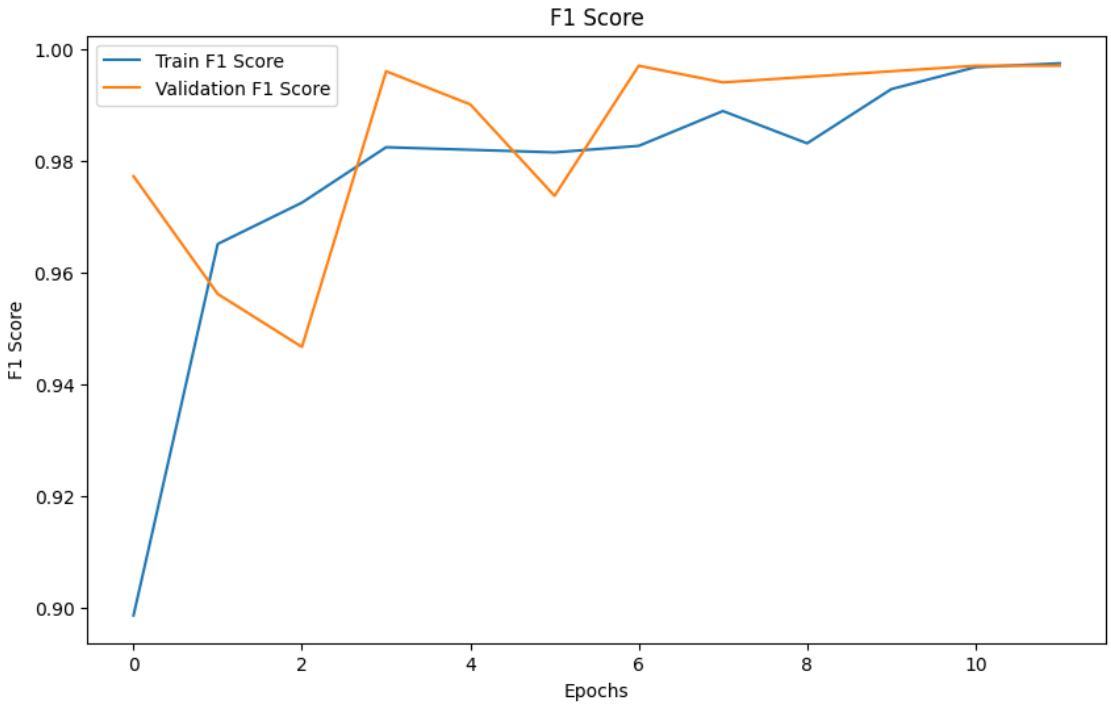
```
#  Precision
plt.figure(figsize=(10, 6))
plt.plot(model_history.history['precision'], label='Train Precision')
plt.plot(model_history.history['val_precision'], label='Validation Precision')
plt.title('Precision')
plt.xlabel('Epochs')
plt.ylabel('Precision')
plt.legend()
plt.savefig('PrecisionVal_precision.png')
plt.show()

# Recall
plt.figure(figsize=(10, 6))
plt.plot(model_history.history['recall'], label='Train Recall')
plt.plot(model_history.history['val_recall'], label='Validation Recall')
plt.title('Recall')
plt.xlabel('Epochs')
plt.ylabel('Recall')
plt.legend()
plt.savefig('RecallVal_recall.png')
plt.show()

# F1-score
plt.figure(figsize=(10, 6))
plt.plot(model_history.history['f1_score'], label='Train F1 Score')
plt.plot(model_history.history['val_f1_score'], label='Validation F1 Score')
plt.title('F1 Score')
plt.xlabel('Epochs')
plt.ylabel('F1 Score')
plt.legend()
plt.savefig('F1ScoreVal_f1_score.png')
plt.show()
```


Loss
Train Loss
Validation Loss
Loss
Epochs
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
2
4
6
8
10


Accuracy
Train Accuracy
Validation Accuracy
Accuracy
Epochs
0.90
0.92
0.94
0.96
0.98
1.00
0
2
4
6
8
10


Precision
Train Precision
Validation Precision
Precision
Epochs
1.00
0.98
0.96
0.94
0.92
0.90
0
2
4
6
8
10


Recall
Train Recall
Validation Recall
Recall
Epochs
1.00
0.98
0.96
0.94
0.92
0.90
0
2
4
6
8
10



0.15 Save Model

[22]:
```
model.save('/content/drive/MyDrive/Car_image_classification_CNN_model/Models/
↪transfer_learning_model_20_epoch.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

[32]:
```
validation_set = val_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/val',

    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
↪'3_4th_driver_side_front', '3_4th_passenger_side_front',]
)


class_labels = {v: k for k, v in validation_set.class_indices.items()}
```

```
plt.figure(figsize=(20, 20))
for i in range(25):
    batch_idx = random.randint(0, len(validation_set) - 1)
    batch = validation_set[batch_idx]
    image_idx = random.randint(0, len(batch[0]) - 1)

    image = batch[0][image_idx]


    image_expanded = np.expand_dims(image, axis=0)
    prediction = model.predict(image_expanded)
    predicted_label_index = np.argmax(prediction, axis=1)[0]
    predicted_label = class_labels[predicted_label_index]

    label = batch[1][image_idx]


    class_index = np.argmax(label)


    if np.max(image) <= 1.0:
        image = (image * 255).astype(np.uint8)

    plt.subplot(5, 5, i + 1)
    plt.imshow(image)
    plt.title(f"True: {class_labels[class_index]}\n Pred: {predicted_label}")
    plt.axis("off")

plt.show()
```

Output hidden; open in https://colab.research.google.com to view.

[27]:
```
test_set = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test',
    target_size=(480, 480),
    batch_size=32,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
)


class_labels = {v: k for k, v in test_set.class_indices.items()}

plt.figure(figsize=(20, 20))
```

```
for i in range(25):
    batch_idx = random.randint(0, len(test_set) - 1)
    batch = test_set[batch_idx]
    image_idx = random.randint(0, len(batch[0]) - 1)

    image = batch[0][image_idx]


    image_expanded = np.expand_dims(image, axis=0)
    prediction = model.predict(image_expanded)
    predicted_label_index = np.argmax(prediction, axis=1)[0]
    predicted_label = class_labels[predicted_label_index]

    label = batch[1][image_idx]


    class_index = np.argmax(label)


    if np.max(image) <= 1.0:
        image = (image * 255).astype(np.uint8)

    plt.subplot(5, 5, i + 1)
    plt.imshow(image)
    plt.title(f"True: {class_labels[class_index]} \n  Pred: {predicted_label}")
    plt.axis("off")

plt.show()
```

Output hidden; open in https://colab.research.google.com to view.

[36]:
```
validation_set = val_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/val',
    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
)


class_labels = {v: k for k, v in validation_set.class_indices.items()}
class_indices = {v: k for k, v in class_labels.items()}

true_labels = []
pred_labels = []
```

```
plt.figure(figsize=(20, 20))
for i in range(25):
    batch_idx = random.randint(0, len(validation_set) - 1)
    batch = validation_set[batch_idx]
    image_idx = random.randint(0, len(batch[0]) - 1)

    image = batch[0][image_idx]


    image_expanded = np.expand_dims(image, axis=0)
    prediction = model.predict(image_expanded)
    predicted_label_index = np.argmax(prediction, axis=1)[0]
    predicted_label = class_labels[predicted_label_index]

    label = batch[1][image_idx]


    true_label_index = np.argmax(label)


    if np.max(image) <= 1.0:
        image = (image * 255).astype(np.uint8)

    true_labels.append(true_label_index)
    pred_labels.append(predicted_label_index)

    plt.subplot(5, 5, i + 1)
    plt.imshow(image)
    plt.title(f"True: {class_labels[true_label_index]}\n Pred:␣
 ↪{predicted_label}")
    plt.axis("off")

plt.show()


conf_matrix = confusion_matrix(true_labels, pred_labels)
disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix,␣
 ↪display_labels=class_labels.values())


disp.plot(cmap=plt.cm.Blues)
plt.xticks(rotation=45, ha='right')
plt.show()
```

Output hidden; open in https://colab.research.google.com to view.

[38]:
```
test_set = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test',
    target_size=(480, 480),
    batch_size=16,
    class_mode='categorical',
    classes=['passenger_side', '3_4th_passenger_side_rear', 'driver_side',␣
 ↪'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
 ↪'3_4th_driver_side_front', '3_4th_passenger_side_front']
)


class_labels = {v: k for k, v in test_set.class_indices.items()}
class_indices = {v: k for k, v in class_labels.items()}

true_labels = []
pred_labels = []

plt.figure(figsize=(20, 20))
for i in range(25):
    batch_idx = random.randint(0, len(test_set) - 1)
    batch = test_set[batch_idx]
    image_idx = random.randint(0, len(batch[0]) - 1)

    image = batch[0][image_idx]


    image_expanded = np.expand_dims(image, axis=0)
    prediction = model.predict(image_expanded)
    predicted_label_index = np.argmax(prediction, axis=1)[0]
    predicted_label = class_labels[predicted_label_index]

    label = batch[1][image_idx]


    true_label_index = np.argmax(label)


    if np.max(image) <= 1.0:
        image = (image * 255).astype(np.uint8)

    true_labels.append(true_label_index)
    pred_labels.append(predicted_label_index)

    plt.subplot(5, 5, i + 1)
    plt.imshow(image)
    plt.title(f"True: {class_labels[true_label_index]} \n Pred:␣
 ↪{predicted_label}")
```

```
    plt.axis("off")

plt.show()


conf_matrix = confusion_matrix(true_labels, pred_labels)
disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix,␣
 ↪display_labels=class_labels.values())


disp.plot(cmap=plt.cm.Blues)
plt.xticks(rotation=45, ha='right')
plt.show()
```

Output hidden; open in https://colab.research.google.com to view.

0.16 Get Model Predictions and Class Names

[39]:
```
y_pred = model.predict(test_set)
y_pred = np.argmax(y_pred, axis=1)
print(y_pred)
class_indices = test_set.class_indices
inverse_class_indices = {v: k for k, v in class_indices.items()}
y_pred_class_names = [inverse_class_indices[idx] for idx in y_pred]
print(y_pred_class_names)
```

```
62/62 [==============================] - 27s 415ms/step
[1 2 1 8 3 3 3 5 5 8 7 7 3 1 7 7 7 2 5 3 1 4 8 7 0 6 0 8 4 8 1 1 2 1 2 0 2
 3 0 1 8 1 1 3 7 7 0 2 8 5 0 3 8 5 4 2 2 5 3 6 0 8 3 1 5 4 2 7 6 0 1 7 6 8
 8 7 4 2 2 0 3 7 2 8 4 5 6 0 2 2 6 6 4 3 8 0 1 6 2 8 8 0 3 5 2 0 4 0 1 2 3
 7 4 6 8 7 2 5 0 0 7 2 5 5 7 5 5 2 6 1 6 3 3 7 7 7 7 3 6 3 7 5 7 4 2 0 0 8
 7 3 6 4 2 0 3 8 3 8 0 4 5 1 2 3 2 4 5 2 5 4 7 7 6 7 3 0 6 8 1 3 4 3 3 2 8
 0 5 1 2 5 4 5 6 1 0 5 6 2 1 6 0 6 0 2 5 1 1 6 0 8 6 0 7 3 7 7 7 7 5 2 2 6
 1 6 6 6 3 4 2 5 3 6 5 5 0 6 1 0 5 1 0 0 6 3 2 1 3 5 2 5 6 4 1 7 5 7 8 1 4
 0 6 1 3 4 6 6 8 7 1 0 2 7 4 8 8 8 1 2 8 1 6 8 1 7 2 7 3 7 1 2 5 1 5 8 2 6
 3 7 5 4 6 3 1 0 5 2 7 6 3 1 8 5 3 1 8 4 8 1 7 7 1 2 4 7 3 1 2 7 2 3 4 8 5
 0 1 3 6 2 3 6 6 4 5 5 7 0 1 7 3 1 0 5 8 6 6 5 1 2 5 4 4 5 1 7 6 8 2 1 1 0
 8 7 7 4 6 3 4 1 0 2 0 5 8 3 4 7 1 5 3 2 2 6 5 7 7 1 6 1 0 3 1 0 4 7 3 4 7
 5 0 7 7 0 1 3 5 6 0 0 0 5 1 2 3 3 5 0 8 4 8 2 1 0 1 3 6 0 8 3 1 0 2 0 1 0
 7 0 2 1 4 5 1 5 0 5 4 8 1 5 8 6 0 0 8 4 6 2 6 1 1 4 8 8 1 4 6 5 0 8 6 4 3
 7 4 1 3 4 8 6 5 0 1 5 0 1 7 4 7 2 0 8 4 4 6 0 5 2 0 3 4 7 6 4 1 1 2 3 1 2
 0 2 8 8 3 3 1 7 5 1 8 7 1 5 7 7 1 4 6 6 7 8 6 3 7 2 5 8 0 6 7 1 8 4 8 8 8
 3 7 4 4 5 5 7 1 5 6 6 2 5 0 0 4 7 5 6 6 0 7 4 3 5 2 6 8 3 8 8 1 1 6 1 5 8
 8 7 2 1 4 2 3 4 1 1 2 4 8 8 5 4 2 0 8 7 1 7 6 7 2 1 3 0 6 5 1 3 0 0 8 3 4
 6 0 8 7 4 1 8 3 3 6 6 7 7 2 6 8 0 1 6 1 8 4 8 3 7 5 8 6 5 8 8 7 7 5 7 1 0
 4 3 7 0 8 4 1 4 0 7 0 7 3 6 4 2 5 2 3 6 7 6 6 5 3 5 3 8 6 7 8 3 3 5 5 3 8
 6 7 6 1 7 6 2 8 3 3 2 3 4 6 5 0 6 8 3 1 0 5 3 3 4 0 4 8 7 7 5 1 8 7 0 4 4
 6 4 6 6 8 6 2 8 5 5 4 2 0 7 0 8 2 6 4 3 8 4 2 0 4 6 3 2 2 7 5 1 3 2 4 1 8
```

```
 7 4 0 8 2 8 0 3 1 5 3 4 3 2 3 5 4 5 5 8 0 6 7 7 3 6 0 8 4 1 2 8 8 5 0 3 1
 4 6 3 8 0 8 5 4 1 5 8 4 1 0 1 8 0 1 5 6 4 6 6 4 4 0 4 8 2 6 2 3 4 3 5 5 0
 2 2 0 1 6 5 6 5 4 2 2 3 5 1 3 8 7 0 7 1 2 5 7 3 0 6 0 5 3 1 0 3 6 7 8 3 2
 8 2 3 8 6 3 8 1 0 6 4 5 0 4 2 7 5 2 5 6 1 3 3 3 1 2 0 5 2 4 2 1 5 0 2 4 8
 7 5 1 0 6 0 3 5 0 5 2 8 4 3 5 0 4 8 8 8 3 2 0 2 5 8 4 6 4 7 4 7 0 8 6 0 7
 1 8 7 0 2 3 7 6 6 4 2 5 2 6 4 2 7 5 7 2 5 3 0 7 5 0 4 0 3]
['3_4th_passenger_side_rear', 'driver_side', '3_4th_passenger_side_rear',
'3_4th_passenger_side_front', 'front', 'front', 'front', 'rear', 'rear',
'3_4th_passenger_side_front', '3_4th_driver_side_front',
'3_4th_driver_side_front', 'front', '3_4th_passenger_side_rear',
'3_4th_driver_side_front', '3_4th_driver_side_front', '3_4th_driver_side_front',
'driver_side', 'rear', 'front', '3_4th_passenger_side_rear', 'unknown',
'3_4th_passenger_side_front', '3_4th_driver_side_front', 'passenger_side',
'3_4th_driver_side_rear', 'passenger_side', '3_4th_passenger_side_front',
'unknown', '3_4th_passenger_side_front', '3_4th_passenger_side_rear',
'3_4th_passenger_side_rear', 'driver_side', '3_4th_passenger_side_rear',
'driver_side', 'passenger_side', 'driver_side', 'front', 'passenger_side',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front',
'3_4th_passenger_side_rear', '3_4th_passenger_side_rear', 'front',
'3_4th_driver_side_front', '3_4th_driver_side_front', 'passenger_side',
'driver_side', '3_4th_passenger_side_front', 'rear', 'passenger_side', 'front',
'3_4th_passenger_side_front', 'rear', 'unknown', 'driver_side', 'driver_side',
'rear', 'front', '3_4th_driver_side_rear', 'passenger_side',
'3_4th_passenger_side_front', 'front', '3_4th_passenger_side_rear', 'rear',
'unknown', 'driver_side', '3_4th_driver_side_front', '3_4th_driver_side_rear',
'passenger_side', '3_4th_passenger_side_rear', '3_4th_driver_side_front',
'3_4th_driver_side_rear', '3_4th_passenger_side_front',
'3_4th_passenger_side_front', '3_4th_driver_side_front', 'unknown',
'driver_side', 'driver_side', 'passenger_side', 'front',
'3_4th_driver_side_front', 'driver_side', '3_4th_passenger_side_front',
'unknown', 'rear', '3_4th_driver_side_rear', 'passenger_side', 'driver_side',
'driver_side', '3_4th_driver_side_rear', '3_4th_driver_side_rear', 'unknown',
'front', '3_4th_passenger_side_front', 'passenger_side',
'3_4th_passenger_side_rear', '3_4th_driver_side_rear', 'driver_side',
'3_4th_passenger_side_front', '3_4th_passenger_side_front', 'passenger_side',
'front', 'rear', 'driver_side', 'passenger_side', 'unknown', 'passenger_side',
'3_4th_passenger_side_rear', 'driver_side', 'front', '3_4th_driver_side_front',
'unknown', '3_4th_driver_side_rear', '3_4th_passenger_side_front',
'3_4th_driver_side_front', 'driver_side', 'rear', 'passenger_side',
'passenger_side', '3_4th_driver_side_front', 'driver_side', 'rear', 'rear',
'3_4th_driver_side_front', 'rear', 'rear', 'driver_side',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear', '3_4th_driver_side_rear',
'front', 'front', '3_4th_driver_side_front', '3_4th_driver_side_front',
'3_4th_driver_side_front', '3_4th_driver_side_front', 'front',
'3_4th_driver_side_rear', 'front', '3_4th_driver_side_front', 'rear',
'3_4th_driver_side_front', 'unknown', 'driver_side', 'passenger_side',
'passenger_side', '3_4th_passenger_side_front', '3_4th_driver_side_front',
'front', '3_4th_driver_side_rear', 'unknown', 'driver_side', 'passenger_side',
```

'front', '3_4th_passenger_side_front', 'front', '3_4th_passenger_side_front',
'passenger_side', 'unknown', 'rear', '3_4th_passenger_side_rear', 'driver_side',
'front', 'driver_side', 'unknown', 'rear', 'driver_side', 'rear', 'unknown',
'3_4th_driver_side_front', '3_4th_driver_side_front', '3_4th_driver_side_rear',
'3_4th_driver_side_front', 'front', 'passenger_side', '3_4th_driver_side_rear',
'3_4th_passenger_side_front', '3_4th_passenger_side_rear', 'front', 'unknown',
'front', 'front', 'driver_side', '3_4th_passenger_side_front', 'passenger_side',
'rear', '3_4th_passenger_side_rear', 'driver_side', 'rear', 'unknown', 'rear',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear', 'passenger_side', 'rear',
'3_4th_driver_side_rear', 'driver_side', '3_4th_passenger_side_rear',
'3_4th_driver_side_rear', 'passenger_side', '3_4th_driver_side_rear',
'passenger_side', 'driver_side', 'rear', '3_4th_passenger_side_rear',
'3_4th_passenger_side_rear', '3_4th_driver_side_rear', 'passenger_side',
'3_4th_passenger_side_front', '3_4th_driver_side_rear', 'passenger_side',
'3_4th_driver_side_front', 'front', '3_4th_driver_side_front',
'3_4th_driver_side_front', '3_4th_driver_side_front', '3_4th_driver_side_front',
'rear', 'driver_side', 'driver_side', '3_4th_driver_side_rear',
'3_4th_passenger_side_rear', '3_4th_driver_side_rear', '3_4th_driver_side_rear',
'3_4th_driver_side_rear', 'front', 'unknown', 'driver_side', 'rear', 'front',
'3_4th_driver_side_rear', 'rear', 'rear', 'passenger_side',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear', 'passenger_side', 'rear',
'3_4th_passenger_side_rear', 'passenger_side', 'passenger_side',
'3_4th_driver_side_rear', 'front', 'driver_side', '3_4th_passenger_side_rear',
'front', 'rear', 'driver_side', 'rear', '3_4th_driver_side_rear', 'unknown',
'3_4th_passenger_side_rear', '3_4th_driver_side_front', 'rear',
'3_4th_driver_side_front', '3_4th_passenger_side_front',
'3_4th_passenger_side_rear', 'unknown', 'passenger_side',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear', 'front', 'unknown',
'3_4th_driver_side_rear', '3_4th_driver_side_rear',
'3_4th_passenger_side_front', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', 'passenger_side', 'driver_side',
'3_4th_driver_side_front', 'unknown', '3_4th_passenger_side_front',
'3_4th_passenger_side_front', '3_4th_passenger_side_front',
'3_4th_passenger_side_rear', 'driver_side', '3_4th_passenger_side_front',
'3_4th_passenger_side_rear', '3_4th_driver_side_rear',
'3_4th_passenger_side_front', '3_4th_passenger_side_rear',
'3_4th_driver_side_front', 'driver_side', '3_4th_driver_side_front', 'front',
'3_4th_driver_side_front', '3_4th_passenger_side_rear', 'driver_side', 'rear',
'3_4th_passenger_side_rear', 'rear', '3_4th_passenger_side_front',
'driver_side', '3_4th_driver_side_rear', 'front', '3_4th_driver_side_front',
'rear', 'unknown', '3_4th_driver_side_rear', 'front',
'3_4th_passenger_side_rear', 'passenger_side', 'rear', 'driver_side',
'3_4th_driver_side_front', '3_4th_driver_side_rear', 'front',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front', 'rear', 'front',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front', 'unknown',
'3_4th_passenger_side_front', '3_4th_passenger_side_rear',
'3_4th_driver_side_front', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', 'driver_side', 'unknown',

'3_4th_driver_side_front', 'front', '3_4th_passenger_side_rear', 'driver_side',
'3_4th_driver_side_front', 'driver_side', 'front', 'unknown',
'3_4th_passenger_side_front', 'rear', 'passenger_side',
'3_4th_passenger_side_rear', 'front', '3_4th_driver_side_rear', 'driver_side',
'front', '3_4th_driver_side_rear', '3_4th_driver_side_rear', 'unknown', 'rear',
'rear', '3_4th_driver_side_front', 'passenger_side',
'3_4th_passenger_side_rear', '3_4th_driver_side_front', 'front',
'3_4th_passenger_side_rear', 'passenger_side', 'rear',
'3_4th_passenger_side_front', '3_4th_driver_side_rear',
'3_4th_driver_side_rear', 'rear', '3_4th_passenger_side_rear', 'driver_side',
'rear', 'unknown', 'unknown', 'rear', '3_4th_passenger_side_rear',
'3_4th_driver_side_front', '3_4th_driver_side_rear',
'3_4th_passenger_side_front', 'driver_side', '3_4th_passenger_side_rear',
'3_4th_passenger_side_rear', 'passenger_side', '3_4th_passenger_side_front',
'3_4th_driver_side_front', '3_4th_driver_side_front', 'unknown',
'3_4th_driver_side_rear', 'front', 'unknown', '3_4th_passenger_side_rear',
'passenger_side', 'driver_side', 'passenger_side', 'rear',
'3_4th_passenger_side_front', 'front', 'unknown', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', 'rear', 'front', 'driver_side', 'driver_side',
'3_4th_driver_side_rear', 'rear', '3_4th_driver_side_front',
'3_4th_driver_side_front', '3_4th_passenger_side_rear',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear', 'passenger_side',
'front', '3_4th_passenger_side_rear', 'passenger_side', 'unknown',
'3_4th_driver_side_front', 'front', 'unknown', '3_4th_driver_side_front',
'rear', 'passenger_side', '3_4th_driver_side_front', '3_4th_driver_side_front',
'passenger_side', '3_4th_passenger_side_rear', 'front', 'rear',
'3_4th_driver_side_rear', 'passenger_side', 'passenger_side', 'passenger_side',
'rear', '3_4th_passenger_side_rear', 'driver_side', 'front', 'front', 'rear',
'passenger_side', '3_4th_passenger_side_front', 'unknown',
'3_4th_passenger_side_front', 'driver_side', '3_4th_passenger_side_rear',
'passenger_side', '3_4th_passenger_side_rear', 'front',
'3_4th_driver_side_rear', 'passenger_side', '3_4th_passenger_side_front',
'front', '3_4th_passenger_side_rear', 'passenger_side', 'driver_side',
'passenger_side', '3_4th_passenger_side_rear', 'passenger_side',
'3_4th_driver_side_front', 'passenger_side', 'driver_side',
'3_4th_passenger_side_rear', 'unknown', 'rear', '3_4th_passenger_side_rear',
'rear', 'passenger_side', 'rear', 'unknown', '3_4th_passenger_side_front',
'3_4th_passenger_side_rear', 'rear', '3_4th_passenger_side_front',
'3_4th_driver_side_rear', 'passenger_side', 'passenger_side',
'3_4th_passenger_side_front', 'unknown', '3_4th_driver_side_rear',
'driver_side', '3_4th_driver_side_rear', '3_4th_passenger_side_rear',
'3_4th_passenger_side_rear', 'unknown', '3_4th_passenger_side_front',
'3_4th_passenger_side_front', '3_4th_passenger_side_rear', 'unknown',
'3_4th_driver_side_rear', 'rear', 'passenger_side',
'3_4th_passenger_side_front', '3_4th_driver_side_rear', 'unknown', 'front',
'3_4th_driver_side_front', 'unknown', '3_4th_passenger_side_rear', 'front',
'unknown', '3_4th_passenger_side_front', '3_4th_driver_side_rear', 'rear',
'passenger_side', '3_4th_passenger_side_rear', 'rear', 'passenger_side',

'3_4th_passenger_side_rear', '3_4th_driver_side_front', 'unknown',
'3_4th_driver_side_front', 'driver_side', 'passenger_side',
'3_4th_passenger_side_front', 'unknown', 'unknown', '3_4th_driver_side_rear',
'passenger_side', 'rear', 'driver_side', 'passenger_side', 'front', 'unknown',
'3_4th_driver_side_front', '3_4th_driver_side_rear', 'unknown',
'3_4th_passenger_side_rear', '3_4th_passenger_side_rear', 'driver_side',
'front', '3_4th_passenger_side_rear', 'driver_side', 'passenger_side',
'driver_side', '3_4th_passenger_side_front', '3_4th_passenger_side_front',
'front', 'front', '3_4th_passenger_side_rear', '3_4th_driver_side_front',
'rear', '3_4th_passenger_side_rear', '3_4th_passenger_side_front',
'3_4th_driver_side_front', '3_4th_passenger_side_rear', 'rear',
'3_4th_driver_side_front', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', 'unknown', '3_4th_driver_side_rear',
'3_4th_driver_side_rear', '3_4th_driver_side_front',
'3_4th_passenger_side_front', '3_4th_driver_side_rear', 'front',
'3_4th_driver_side_front', 'driver_side', 'rear', '3_4th_passenger_side_front',
'passenger_side', '3_4th_driver_side_rear', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front', 'unknown',
'3_4th_passenger_side_front', '3_4th_passenger_side_front',
'3_4th_passenger_side_front', 'front', '3_4th_driver_side_front', 'unknown',
'unknown', 'rear', 'rear', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', 'rear', '3_4th_driver_side_rear',
'3_4th_driver_side_rear', 'driver_side', 'rear', 'passenger_side',
'passenger_side', 'unknown', '3_4th_driver_side_front', 'rear',
'3_4th_driver_side_rear', '3_4th_driver_side_rear', 'passenger_side',
'3_4th_driver_side_front', 'unknown', 'front', 'rear', 'driver_side',
'3_4th_driver_side_rear', '3_4th_passenger_side_front', 'front',
'3_4th_passenger_side_front', '3_4th_passenger_side_front',
'3_4th_passenger_side_rear', '3_4th_passenger_side_rear',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear', 'rear',
'3_4th_passenger_side_front', '3_4th_passenger_side_front',
'3_4th_driver_side_front', 'driver_side', '3_4th_passenger_side_rear',
'unknown', 'driver_side', 'front', 'unknown', '3_4th_passenger_side_rear',
'3_4th_passenger_side_rear', 'driver_side', 'unknown',
'3_4th_passenger_side_front', '3_4th_passenger_side_front', 'rear', 'unknown',
'driver_side', 'passenger_side', '3_4th_passenger_side_front',
'3_4th_driver_side_front', '3_4th_passenger_side_rear',
'3_4th_driver_side_front', '3_4th_driver_side_rear', '3_4th_driver_side_front',
'driver_side', '3_4th_passenger_side_rear', 'front', 'passenger_side',
'3_4th_driver_side_rear', 'rear', '3_4th_passenger_side_rear', 'front',
'passenger_side', 'passenger_side', '3_4th_passenger_side_front', 'front',
'unknown', '3_4th_driver_side_rear', 'passenger_side',
'3_4th_passenger_side_front', '3_4th_driver_side_front', 'unknown',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front', 'front', 'front',
'3_4th_driver_side_rear', '3_4th_driver_side_rear', '3_4th_driver_side_front',
'3_4th_driver_side_front', 'driver_side', '3_4th_driver_side_rear',
'3_4th_passenger_side_front', 'passenger_side', '3_4th_passenger_side_rear',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear',

'3_4th_passenger_side_front', 'unknown', '3_4th_passenger_side_front', 'front',
'3_4th_driver_side_front', 'rear', '3_4th_passenger_side_front',
'3_4th_driver_side_rear', 'rear', '3_4th_passenger_side_front',
'3_4th_passenger_side_front', '3_4th_driver_side_front',
'3_4th_driver_side_front', 'rear', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', 'passenger_side', 'unknown', 'front',
'3_4th_driver_side_front', 'passenger_side', '3_4th_passenger_side_front',
'unknown', '3_4th_passenger_side_rear', 'unknown', 'passenger_side',
'3_4th_driver_side_front', 'passenger_side', '3_4th_driver_side_front', 'front',
'3_4th_driver_side_rear', 'unknown', 'driver_side', 'rear', 'driver_side',
'front', '3_4th_driver_side_rear', '3_4th_driver_side_front',
'3_4th_driver_side_rear', '3_4th_driver_side_rear', 'rear', 'front', 'rear',
'front', '3_4th_passenger_side_front', '3_4th_driver_side_rear',
'3_4th_driver_side_front', '3_4th_passenger_side_front', 'front', 'front',
'rear', 'rear', 'front', '3_4th_passenger_side_front', '3_4th_driver_side_rear',
'3_4th_driver_side_front', '3_4th_driver_side_rear',
'3_4th_passenger_side_rear', '3_4th_driver_side_front',
'3_4th_driver_side_rear', 'driver_side', '3_4th_passenger_side_front', 'front',
'front', 'driver_side', 'front', 'unknown', '3_4th_driver_side_rear', 'rear',
'passenger_side', '3_4th_driver_side_rear', '3_4th_passenger_side_front',
'front', '3_4th_passenger_side_rear', 'passenger_side', 'rear', 'front',
'front', 'unknown', 'passenger_side', 'unknown', '3_4th_passenger_side_front',
'3_4th_driver_side_front', '3_4th_driver_side_front', 'rear',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front',
'3_4th_driver_side_front', 'passenger_side', 'unknown', 'unknown',
'3_4th_driver_side_rear', 'unknown', '3_4th_driver_side_rear',
'3_4th_driver_side_rear', '3_4th_passenger_side_front',
'3_4th_driver_side_rear', 'driver_side', '3_4th_passenger_side_front', 'rear',
'rear', 'unknown', 'driver_side', 'passenger_side', '3_4th_driver_side_front',
'passenger_side', '3_4th_passenger_side_front', 'driver_side',
'3_4th_driver_side_rear', 'unknown', 'front', '3_4th_passenger_side_front',
'unknown', 'driver_side', 'passenger_side', 'unknown', '3_4th_driver_side_rear',
'front', 'driver_side', 'driver_side', '3_4th_driver_side_front', 'rear',
'3_4th_passenger_side_rear', 'front', 'driver_side', 'unknown',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front',
'3_4th_driver_side_front', 'unknown', 'passenger_side',
'3_4th_passenger_side_front', 'driver_side', '3_4th_passenger_side_front',
'passenger_side', 'front', '3_4th_passenger_side_rear', 'rear', 'front',
'unknown', 'front', 'driver_side', 'front', 'rear', 'unknown', 'rear', 'rear',
'3_4th_passenger_side_front', 'passenger_side', '3_4th_driver_side_rear',
'3_4th_driver_side_front', '3_4th_driver_side_front', 'front',
'3_4th_driver_side_rear', 'passenger_side', '3_4th_passenger_side_front',
'unknown', '3_4th_passenger_side_rear', 'driver_side',
'3_4th_passenger_side_front', '3_4th_passenger_side_front', 'rear',
'passenger_side', 'front', '3_4th_passenger_side_rear', 'unknown',
'3_4th_driver_side_rear', 'front', '3_4th_passenger_side_front',
'passenger_side', '3_4th_passenger_side_front', 'rear', 'unknown',
'3_4th_passenger_side_rear', 'rear', '3_4th_passenger_side_front', 'unknown',

```
'3_4th_passenger_side_rear', 'passenger_side', '3_4th_passenger_side_rear',
'3_4th_passenger_side_front', 'passenger_side', '3_4th_passenger_side_rear',
'rear', '3_4th_driver_side_rear', 'unknown', '3_4th_driver_side_rear',
'3_4th_driver_side_rear', 'unknown', 'unknown', 'passenger_side', 'unknown',
'3_4th_passenger_side_front', 'driver_side', '3_4th_driver_side_rear',
'driver_side', 'front', 'unknown', 'front', 'rear', 'rear', 'passenger_side',
'driver_side', 'driver_side', 'passenger_side', '3_4th_passenger_side_rear',
'3_4th_driver_side_rear', 'rear', '3_4th_driver_side_rear', 'rear', 'unknown',
'driver_side', 'driver_side', 'front', 'rear', '3_4th_passenger_side_rear',
'front', '3_4th_passenger_side_front', '3_4th_driver_side_front',
'passenger_side', '3_4th_driver_side_front', '3_4th_passenger_side_rear',
'driver_side', 'rear', '3_4th_driver_side_front', 'front', 'passenger_side',
'3_4th_driver_side_rear', 'passenger_side', 'rear', 'front',
'3_4th_passenger_side_rear', 'passenger_side', 'front',
'3_4th_driver_side_rear', '3_4th_driver_side_front',
'3_4th_passenger_side_front', 'front', 'driver_side',
'3_4th_passenger_side_front', 'driver_side', 'front',
'3_4th_passenger_side_front', '3_4th_driver_side_rear', 'front',
'3_4th_passenger_side_front', '3_4th_passenger_side_rear', 'passenger_side',
'3_4th_driver_side_rear', 'unknown', 'rear', 'passenger_side', 'unknown',
'driver_side', '3_4th_driver_side_front', 'rear', 'driver_side', 'rear',
'3_4th_driver_side_rear', '3_4th_passenger_side_rear', 'front', 'front',
'front', '3_4th_passenger_side_rear', 'driver_side', 'passenger_side', 'rear',
'driver_side', 'unknown', 'driver_side', '3_4th_passenger_side_rear', 'rear',
'passenger_side', 'driver_side', 'unknown', '3_4th_passenger_side_front',
'3_4th_driver_side_front', 'rear', '3_4th_passenger_side_rear',
'passenger_side', '3_4th_driver_side_rear', 'passenger_side', 'front', 'rear',
'passenger_side', 'rear', 'driver_side', '3_4th_passenger_side_front',
'unknown', 'front', 'rear', 'passenger_side', 'unknown',
'3_4th_passenger_side_front', '3_4th_passenger_side_front',
'3_4th_passenger_side_front', 'front', 'driver_side', 'passenger_side',
'driver_side', 'rear', '3_4th_passenger_side_front', 'unknown',
'3_4th_driver_side_rear', 'unknown', '3_4th_driver_side_front', 'unknown',
'3_4th_driver_side_front', 'passenger_side', '3_4th_passenger_side_front',
'3_4th_driver_side_rear', 'passenger_side', '3_4th_driver_side_front',
'3_4th_passenger_side_rear', '3_4th_passenger_side_front',
'3_4th_driver_side_front', 'passenger_side', 'driver_side', 'front',
'3_4th_driver_side_front', '3_4th_driver_side_rear', '3_4th_driver_side_rear',
'unknown', 'driver_side', 'rear', 'driver_side', '3_4th_driver_side_rear',
'unknown', 'driver_side', '3_4th_driver_side_front', 'rear',
'3_4th_driver_side_front', 'driver_side', 'rear', 'front', 'passenger_side',
'3_4th_driver_side_front', 'rear', 'passenger_side', 'unknown',
'passenger_side', 'front']
```

0.17 Load Pre-trained Model

[40]:
```
model = load_model('/content/drive/MyDrive/Car_image_classification_CNN_model/
↪Models/transfer_learning_model_20_epoch.h5', compile=True)
```

0.18 Predict Images and Display Results

[41]:
```
def predict_images(model, image_paths):
    class_labels = ['passenger_side', '3_4th_passenger_side_rear',␣
↪'driver_side', 'front', 'unknown', 'rear', '3_4th_driver_side_rear',␣
↪'3_4th_driver_side_front', '3_4th_passenger_side_front']

    for img_path in image_paths:

        img = image.load_img(img_path, target_size=(480, 480))


        x = image.img_to_array(img)
        x = x / 255
        x = np.expand_dims(x, axis=0)


        prediction = model.predict(x)


        predicted_class_index = np.argmax(prediction)
        predicted_class_label = class_labels[predicted_class_index]

        plt.imshow(img)
        plt.title(f'Predicted class: {predicted_class_label}')
        plt.axis('off')
        plt.show()
```

[42]:
```
image_paths = [
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test/
↪front/102158.jpg',
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test/
↪unknown/163304.jpg',
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test/
↪3_4th_passenger_side_front/28330.jpg',
    '/content/drive/MyDrive/Car_image_classification_CNN_model/split_data/test/
↪3_4th_driver_side_rear/34327.jpg',]
```

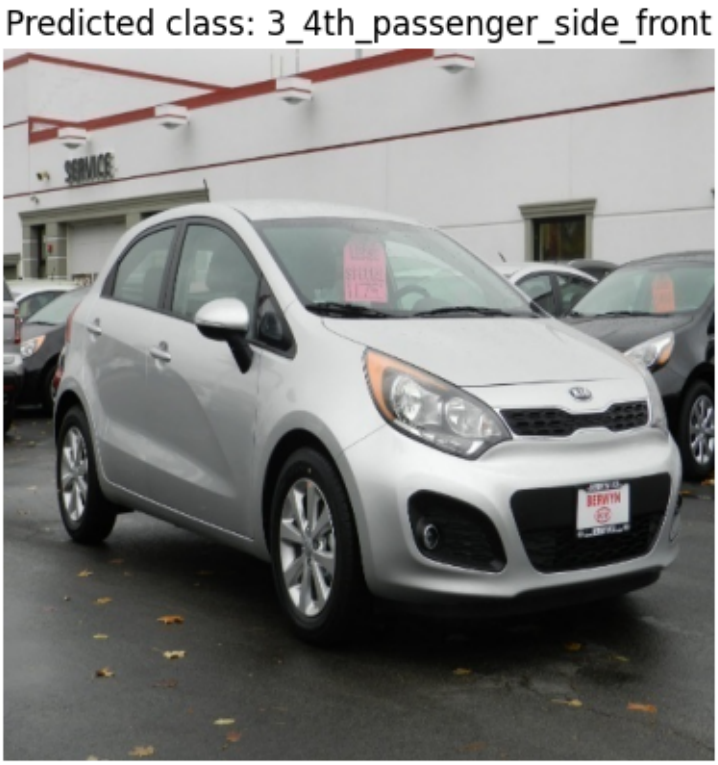
[44]:
```
# Function Call
predict_images(model, image_paths)
```

```
1/1 [==============================] - 2s 2s/step
```

Predicted class: front


```
1/1 [==============================] - 0s 27ms/step
```


```
1/1 [==============================] - 0s 35ms/step
```




```
1/1 [==============================] - 0s 26ms/step
```

[7]:
```
!jupyter nbconvert --to html "/content/drive/MyDrive/
  ↪Car_image_classification_CNN_model/NoteBook/Model_Transfer_learning.ipynb"
!apt-get install texlive-xetex texlive-fonts-recommended texlive-plain-generic
!jupyter nbconvert --to pdf "/content/drive/MyDrive/
  ↪Car_image_classification_CNN_model/NoteBook/Model_Transfer_learning.ipynb"
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Car_image_classificati
on_CNN_model/NoteBook/Model_Transfer_learning.ipynb to pdf
[NbConvertApp] Support files will be in Model_Transfer_learning_files/
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Making directory ./Model_Transfer_learning_files
[NbConvertApp] Writing 168403 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
```

```
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 951013 bytes to /content/drive/MyDrive/Car_image_classifi
cation_CNN_model/NoteBook/Model_Transfer_learning.pdf
```

[ ]: